PATENT ABSTRACTS OF JAPAN

(11)Publication number : **10-222510**

(43)Date of publication of application : **21.08.1998**

(51)Int.CI. **G06F 17/27**

(21)Application number : **09-024811**

(22)Date of filing : **07.02.1997**

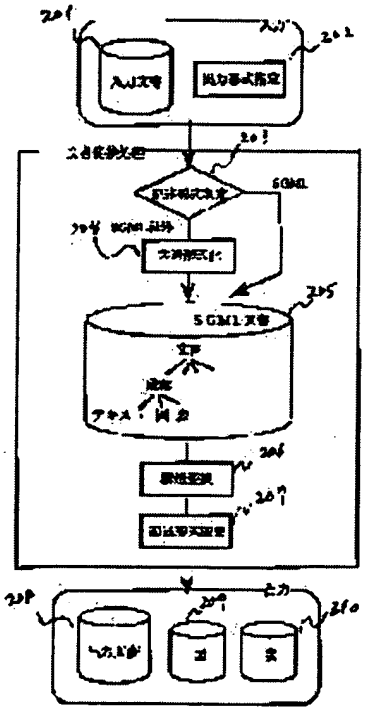
(71)Applicant : **HITACHI LTD**

(72)Inventor : **TAKITA YUKIE**

TAKAHASHI TORU

ITO YASUKI

(54) DOCUMENT CONVERTING METHOD



(57) Abstract:

PROBLEM TO BE SOLVED: To provide a converting method which converts a document that includes a figure created by a word processor, etc., into a document in a format that matches a user's document preparing/referring environment.

SOLUTION: Descriptive format decision 203 that decides whether an input document 201 is described in an SGML (document description language) is performed. Except the case of the SGML, an SGML document 205 is created by executing common format that follows the syntax of the SGML, and when a figure is included in a document, files 209 and 210 are created for each figure through syntax conversion 206 and changed (207) into a desired description format.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the document processing system program which operates on a computer apparatus, and relates to the document processing system approach of performing the structural transition of a document, and conversion of a symbolic convention, especially about the document drawn up with a word processor etc.

[0002]

[Description of the Prior Art] The electronization of a document progressed by the spread of word processors, and it became reusable [the document of editing the document drawn up in the past and drawing up a new document]. However, since the word processor of various models existed and each model used the respectively original document-description format, exchange of the document data between different models was difficult. Although it was available from all models when it was the document of the format of only a simple text, exchange/playback of a document including a graph or a document including layout assignment were not completed.

[0003] The standard document-description language SGML for expressing the logical structure of a document (ISO 8879, Information processing-Text and office systems-Standard Generalized Markup Language (SGML)) was proposed that this problem should be solved. DTD (Document Type Definition: document type definition) defines the set of the structure element which constitutes the structure of a document, and it, and SGML describes a document by it based on this. By surrounding with a tag shows clearly the structure element which constitutes a document. For example, the description of "being a </title> about the <title> conversion approach" expresses that the title of a document is the "conversion approach." What surrounded what surrounded the structure element name (this example "title") by "<" and ">" by the initiation tag, the call, "</", and ">" is called a termination tag.

[0004] As long as it is data according to a symbolic convention of a text expression like PostScript data, you may describe including a graph in a document. Since you cannot describe the image data of a binary format etc. directly in a document, refer to the file in which image data was stored for it using the functor of "entity declaration." In any case, "NOTESHON declaration" shows the symbolic convention of drawing or a table.
[0005] Moreover, the layout information about the layout of a document is not included in a document. In the system which performs layout processing of a document, the layout of a document is performed by associating the structure element and the layout. Therefore, creation of the document independent of the device which draws up a document is attained, and it becomes reusable [a document including a graph].
[0006]
[Problem(s) to be Solved by the Invention] Before introducing SGML, in order to build the system which enables reuse of a document also including the document drawn up with the word processor etc., it is necessary to change into a standard SGML document the document described in the document-description format which changes with word processors of various models etc., respectively. Furthermore, for the broad activity of a document, conversion in other formats [document / SGML] is also needed.
[0007] The purpose of this invention is to offer the conversion approach which is not restricted to an SGML document of changing into the document of the format suitable for a user's document preparation / reference environment a document including the graph created with the word processor etc.
[0008]
[Means for Solving the Problem] In the document drawn up with the word processor etc., the character string showing graph data besides the character string showing the contents of a document and the specific character string showing a character string and the layout information about a diagrammatic display are described by the document-description format defined uniquely. If the specific character string contained in such a word processor document is transposed to the tag expression of SGML, formally, the document according to the functor of SGML is generable.
[0009] By the way, in invention given in JP,7-105216,A, after analyzing document structure by making an SGML document into an input-statement document, a means by which a user can specify easily the processing for performing character string conversion of a structure element unit and the structural transition of document structure is offered by having a means to perform processing corresponding to each structure element which constitutes document structure. And conversion of an SGML document is realized by performing processing specified about each structure element, following document

structure. Therefore, it becomes convertible [the document using the above-mentioned method] by considering that the document of the SGML description generated from a word processor document is an SGML document.

[0010] Moreover, when a graph is included in a document, it is also necessary to extract drawing data as an image file, or for other applications to change and extract tabular data in an available format. By transposing a word processor document to SGML description, the specific character string which surely exists in the head and tail of graph data which are contained in it is transposed to a tag with the tag name showing drawing data or tabular data. It can be considered that the part surrounded with the tag showing drawing data is the structure element of drawing. Drawing data can be cut down by performing about this processing which outputs the contents to another file. If required, this will be changed into a binary format and an image file will be generated. It can be considered that the part which similarly was surrounded with the tag showing tabular data is the structure element of a table. About tabular data, since the specific character string which shows the information about the structure of tables, such as a ruled line location, is also transposed to a tag, it is expressed as front structure data with which the tabular data itself consists of structure elements, such as ruled line information. Therefore, since the processing which should be performed also about each structure element, such as ruled line information, can be defined, it is also easy to generate the table which stores all the information about the structure of a table, and to grasp the structure of a table. Moreover, the tabular data according to the symbolic convention of a request of a user is also generable by performing structure transform processing and character string transform processing of a structure element unit.

[0011]

[Embodiment of the Invention] Hereafter, the example of this invention is explained based on a drawing.

[0012] Drawing 1 shows the system configuration which placed the document transform-processing program which changes a document on the computer connected to the network as an example using the document conversion method of this invention of a system configuration. The computer 1 connected to the network 7 consists of data files 6 for saving the document acquired from other computers through the document and network 7 which are inputted as a display 2, the data entry units 3, such as a keyboard, CPU4, and memory 5 from a data entry unit 3. The common format-ized program 5-2 started by memory 5 from the document transform-processing program 5-1 and the document transform-processing program 5-1, The document structural-analysis program 5-3 started from the document transform-processing program 5-1, The document

structure storing field 5-4 for storing the document structure data with which an SGML document is read, and the document structural-analysis program 5-3 carries out structural analysis of this, and generates it, The image transformation processing program 5-7 which changes the front structure storing field 5-5 for storing each of the tabular data which the document transform-processing program 5-1 extracts from the SGML document with which a graph is included, or image data, the image storing field 5-6, and data format of an image file is placed.

[0013] Drawing 2 shows the outline of document transform processing. An input-statement document is taken as the document stored in portable mold media, such as the document and floppy disk which were created on the computer 1, and CD-ROM, or the document acquired through the network 7. A user specifies the symbolic convention of an output-statement document at the time of a document input. In document transform processing, the symbolic convention of an input-statement document is judged first. If an input-statement document is an SGML document, the document of the specified description will be generated by generating the document structure data of the shape of the tree structure as shown all over drawing, and performing structural transition and symbolic-convention conversion to this document structure data. About documents other than an SGML document, it changes into the common formal document of SGML description first. It is possible to transpose the instruction statement which specifies the layout of centering of a character string included as an approach for changing into SGML description in the document drawn up with the word processor etc. to a tag expression. Structural transition and symbolic-convention conversion are performed like an SGML document by generating a common formal document by such approach, and considering that this is an SGML document. Since the document structure divided into the part of character strings other than a graph, the part of drawing, and the part of a table is generated when a graph is included in a document, it is also possible to output only the part of drawing as an image file, or to output only the part of a table as a front data file.

[0014] Drawing 3 shows the flow chart of a document transform-processing program. The symbolic convention of an input-statement document is judged at step 301. A symbolic convention can be easily judged by referring to a head part, as for many of documents drawn up with a word processor etc., since the symbolic convention is specified by the head part of document data. When an input-statement document is not an SGML document as a result of a judgment, common format-ized processing in which document data are changed into the common formal document of SGML description at step 303 is performed. The text document described according to the functor of LATEX shown in drawing 4 is explained to a detail about common format-ized processing of step 303 as an

example of an input-statement document. A LATEX document starts in ¥documentstyle {...} and consists of a character string showing the contents of a document, and instruction statement about the layout of a document. The instruction statement (for example, ¥title) which starts in ¥ is connected with layout information, such as arrangement for arranging a document, a font, and a character size. except for special instruction statement (for example, ¥documentstyle {jreport}), the appointed layout is applied by LATEX to the character string enclosed in the braces ({--}) which continues after instruction statement. Although there is a character string which is not enclosed in instruction statement and a braces, i.e., the character string to which instruction statement is not given, in such a case, the layout using standard arrangement, a font, and a character size is applied. In common format-ized processing, the expression of instruction statement is transposed to the expression of a tag about a document like drawing 4 . for example, -- a title -- a part -- "-- ¥ -- title -- {-- ODA -- having been based --} -- " -- ******** -- "-- ¥ -- title -- {-- " -- < -- title -- > -- replacing -- after that -- a character string -- continuing -- "--} -- " -- < -- /-- title -- > -- replacing -- things -- "-- < -- title>ODA -- having been based -- < -- /-- title -- > -- " -- ** -- saying -- description -- generating . the same -- a chapter -- a title -- a part -- "-- ¥ -- chapter -- {--} -- " -- a knot -- a title -- a part -- "-- ¥ -- section -- {--} -- " -- respectively -- "-- < -- chapter -- > -- < -- /-- chapter -- > -- " -- "-- < -- section -- > -- < -- /-- section -- > -- " -- replacing . About the character string to which instruction statement which appears after a chapter title or a knot title is not given, it considers that this is a paragraph and the tag <para> showing a paragraph is added to the head and tail of a character string. Moreover, the tag (<doc>) which expresses initiation of a document and termination to the document itself is added to the head of a document, and a tail. By performing such processing, a common formal document like drawing 5 R> 5 is obtained.

[0015] Next, at step 304, it considers that the common formal document generated at the inputted SGML document or step 303 is an SGML document, document syntax analysis is performed, and tree structure-like document structure data are generated. In processing of this step 304, document structure data as shown in drawing 6 are generated by making a common formal document as shown in drawing 5 into an input-statement document. Processing after 305 steps shall be performed about the document structure data generated at 304 steps, and a method given in JP,7-105216,A shall perform assignment of transform processing about each structure element contained in document structure, and activation of those transform processing.

[0016] In step 305, when it judges whether drawing is included or not and drawing is included in a document, image data generation processing in which the part of drawing is

extracted as an image data file is performed at step 306. In step 307, when it judges whether a table is included or not and a table is included in a document, at step 308, the structure of a table is analyzed and tabular data generation processing which generates description of the table according to the specified output form is performed.

[0017] At step 309, since the document of the appointed format is outputted, the structural transition of document structure data, such as replacing the sequence of removal and a structure element for a specific structure element, is performed. For example, since the layout information included in a common formal document becomes unnecessary in changing a common formal document like drawing 5 into an SGML document, the structure element about layout information is removed from document structure data as shown in drawing 6 , a structure element name is changed if needed, and it changes into hierarchical document structure like the "report" shown in drawing 7 .

[0018] At step 310, the document of the appointed output form is generated by changing and outputting to the symbolic convention which had the character string specified about the document structure data after structural transition. For example, an SGML document like drawing 8 can be outputted by repeating a character string output according to the contents of each structure element, following document structure data like drawing 7 .

[0019] Drawing 9 shows the flow of the image data generation processing shown in step 306 of drawing 3 . In SGML, there are an approach of describing in a document only the file name of the image data file (it considers as an image file hereafter.) which exists on a data file 6 as the description approach of the drawing data (image data) contained in a document, and the approach of describing in a document the image data of text format which changed the image data of a binary format into the text expression. In image data generation processing, about the document with which the image data of text format is described in the document, the image data embedded into the document is extracted as an image file, and the image file name is written in into a document. Therefore, it is not necessary to be image data generation processing about that the image file name is described to be in the document from the first. Hereafter, image data generation processing is explained to a detail. The specific character string which shows initiation of image data and termination exists in the head and tail of image data which are contained in a document. Therefore, if the document containing the image data of the bit map format changed into the text expression is changed into a common format, description like drawing 10 will be obtained. If the tag showing drawing is set to <PICTURE>, the bit map data changed into the text expression will be surrounded with the <PICTURE> tag and a </PICTURE> tag. By carrying out structural analysis of such description, document structure data with a bit map data character string are generated as a child of a PICTURE

structure element like drawing 11 . A data storage format is easily acquired by the data storage format's being described by the head of image data as information (it considering as image header information hereafter.) about image data, therefore generally, reading image header information. So, at step 3062, the header information included in the bit map data character string changed into the text expression is read, and the data storage format of drawing is acquired. At step 3064, the file name for storing only drawing data is generated. It outputs as a text file with the file name generated at step 3064 by step 3066 in the bit map data character string which is the child of drawing (PICTURE). At step 3068, the child (bit map data character string) of drawing (PICTURE) is transposed to the file name generated at step 3064. At step 3070, data format is changed about the drawing file outputted at step 3066. For example, binary conversion is performed about the text file outputted at step 3066 about text-ized bit map data like drawing 10 , and a bitmap file is generated. Furthermore, conversion in other image data storage formats is performed using the image transformation program 5-6 if needed.

[0020] Drawing 12 shows the flow of the tabular data generation processing shown in step 308 of drawing 3 . The example of the table set as the object of tabular data generation processing is shown in drawing 13 . If LATEX describes the table of drawing 13 , it can describe like drawing 14 . Hereafter, description of drawing 1414 is explained. The first ¥begin {tabular} expresses initiation of front description. It specifies that are the parameter which specifies the attribute of the line of a table, one line consists of three cels, and {|c|c|c|} following it centers a character string for between each cel in a break and each cel by the vertical ruled line. It can mean that ¥hline and ¥cline draw a horizontal ruled line in that location, ¥hline can draw a ruled line in all the cels contained in a line, and ¥cline {2-3} can specify the range of the cel which draws a ruled line with a parameter (this example {2-3}). Moreover, & expresses the break location between the cels of a table, and ¥¥ expresses line feed. The last ¥end{tabular} expresses termination of front description.

[0021] Front description as shown in drawing 14 is changed into a common format like drawing 15 by common format-ized processing shown in step 303 of drawing 3 . Document structure data like drawing 16 are generated by performing document structural-analysis processing shown in step 304 of drawing 3 about description of drawing 15 . In tabular data generation processing, processing which generates grasp of front structure and desired front description is performed for document structure data like drawing 16 . First, at step 3082, a front structure table as shown in drawing 17 based on this document structure data is generated, and the character string in the ruled line information about all the cels contained in a table and a cel is written in the front structure

table. The number of integrated cels shall be storable in the ruled line information in a table about integration of the cel which adjoins the existence, its position coordinate, lengthwise direction, or longitudinal direction of a ruled line of the four directions surrounding a cel as information required in order to detect integration of a cel.

[0022] About document structure data like drawing 16 , the ruled line information on each cel is written in based on the structure elements hline and cline about the ruled line of a table. "hline" draws the bottom ruled line of all the cels contained in the line, and it serves as an upper ruled line of all the cels contained in coincidence at the following line. In order that "cline" (for example, suppose that it has an attribute "2-3".) may draw a bottom ruled line only in the 2 or 3rd cel of the line, an upper ruled line will exist only in the 2 or 3rd cel also about the following line.

[0023] Integration of the lengthwise direction between cels / longitudinal direction is detected based on such ruled line information after write-in termination of the ruled line information on all cels. If the table of drawing 13 is taken for an example, the cel at the left end of eye one train and the cel at the left end of eye two trains are unified by the lengthwise direction. That is, the cel at the left end of eye one train is an integrated initiation cel of a lengthwise direction, and this can be detected in a front structure table as a cel in which an upper ruled line exists and a bottom ruled line does not exist. If the integrated initiation cel of a lengthwise direction is detected, if there is a bottom ruled line with reference to the ruled line information on the cel which adjoins the lengthwise direction of the following train, it will be made into an integrated termination cel, and if there is no bottom ruled line, it will be regarded as what integration of a lengthwise direction follows further. Integration of the cel of a lengthwise direction follows an adjacent cel in order, it continues until it arrives at the cel in which a bottom ruled line exists, and it uses as an integrated termination cel the cel in which a bottom ruled line exists. The number of the cels from an integrated initiation cel to an integrated termination cel is written in as the number of lengthwise direction integration in the ruled line information about an integrated initiation cel. By paying one's attention to the right ruled line of a cel similarly about lateral integration, the number of integration is detected and it writes in as the number of longitudinal direction integration. However, in LATEX, since the number of integration can be described as a parameter of instruction statement ¥multicolumn about integration of the longitudinal direction of a cel, the number of integration is easily obtained from a parameter. A table like drawing 18 is described by LATEX like drawing 19 .

[0024] If front structure table generation of step 3082 is completed, the front structural transition which changes front structure like drawing 16 into the structure doubled with the output form at step 3084 will be performed. For example, in outputting the document

of a HTML format, it changes into structure like drawing 20 suitable for the functor of HTML. At step 3086, HTML description like drawing 21 is outputted by outputting a character string, following such the tree structure.

[0025]

[Effect of the Invention] According to this invention, the document of the common format of SGML description is generated by transposing the specific character string showing layout information to the tag expression of SGML about a word processor document. By considering that this is an SGML document and processing it, modification of the symbolic convention of a document and various conversion of the document of taking out the graph included in a document can be easily performed now. Therefore, it becomes exchange of the document which does not ask a model, and reusable.

CLAIMS

[Claim(s)]

[Claim 1] A means to transpose the specific character string showing the layout information included in a document to the tag expression of SGML, and to generate the document of the common format of SGML description, A means to analyze the document structure about the SGML document of arbitration, and a means to specify the processing which should be performed to the structure element of the arbitration which constitutes this SGML document, In the document inverter equipped with a means to perform processing specified as each structure element The document of the common format which permuted the specific character string showing layout information by the tag expression of an SGML format about said document is generated. The document conversion approach of changing a document by performing processing which analyzed document structure like the common SGML document, and was beforehand specified about each structure element in this common formal document.

[Claim 2] The document conversion approach of generating an image file and changing data format of an image file by changing said drawing data into a binary format according to the document format of a conversion place if a part for said drawing data division is started, it outputs to another file and there is need about the document containing the drawing data by which the text expression is carried out in the document conversion approach according to claim 1 according to the symbolic convention defined beforehand.

[Claim 3] The document conversion approach which generates the front description which has grasped front structure based on tabular data, and followed it at the desired symbolic convention by generating the table which stores the information about the structure of a table about the document containing the tabular data by which the text expression is carried out in the document conversion approach according to claim 1 according to the symbolic convention defined beforehand.

[Claim 4] The document conversion approach which generates the document from which at least one of the document with which the graph created beforehand is included in the document conversion approach according to claim 1 to 3, drawing, and tables was removed.

[Translation done.]

A.B/UPSA/P0211

CITED REFERENCE

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平10－222510

(43)公開日　平成10年(1998)８月21日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　17/27		Ｇ０６Ｆ　15/20	５５０Ｅ

審査請求　未請求　請求項の数４　ＯＬ　（全９頁）

(21)出願番号　特願平9－24811

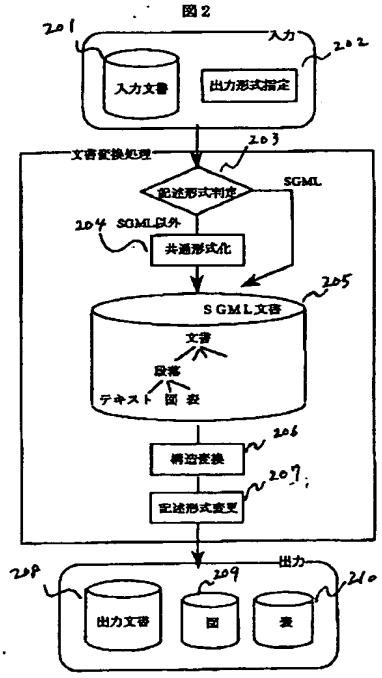
(22)出願日　平成９年(1997)２月７日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　滝田　幸恵
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所情報・通信開発本部内
(72)発明者　高橋　亨
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所情報・通信開発本部内
(72)発明者　伊藤　泰樹
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内
(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　文書変換方法

(57)【要約】
【課題】ワープロ等で作成された図表を含む文書を、ユーザの文書作成／参照環境に合った形式の文書に変換する変換方法を提供する。
【解決手段】入力文書２０１に対してＳＧＭＬで記述されているかを判定する記述形式判定２０３を行い、ＳＧＭＬ以外の場合は、ＳＧＭＬの構文に従った共通形式化２０４を施してＳＧＭＬ文書２０５を作成し、文書に図表が含まれる場合は、構造変換２０６で図表それぞれのファイル２０９、２１０を生成し、所望の記述形式に変更２０７する。

図２



【特許請求の範囲】

【請求項１】文書に含まれるレイアウト情報を表す特定の文字列をＳＧＭＬのタグ表現に置き換え、ＳＧＭＬ記述の共通形式の文書を生成する手段と、任意のＳＧＭＬ文書について、その文書構造を解析する手段と、該ＳＧＭＬ文書を構成する任意の構造要素に対して実行すべき処理を指定する手段と、各構造要素に指定された処理を実行する手段とを備えた文書変換装置において、前記文書について、レイアウト情報を表す特定の文字列をＳＧＭＬ形式のタグ表現に置換した共通形式の文書を生成し、該共通形式文書を一般のＳＧＭＬ文書と同様に、文書構造を解析し、各構造要素についてあらかじめ指定された処理を行うことにより、文書の変換を行う文書変換方法。

【請求項２】請求項１記載の文書変換方法において、予め定められた記述形式に従ってテキスト表現されている図データを含む文書について、前記図データ部分を切り出して別ファイルに出力し、必要があれば、前記図データをバイナリ形式に変換することにより、画像ファイルを生成し、変換先の文書形式に応じて画像ファイルのデータ形式の変換を行う文書変換方法。

【請求項３】請求項１記載の文書変換方法において、予め定められた記述形式に従ってテキスト表現されている表データを含む文書について、表データをもとに、表の構造に関する情報を格納するテーブルを生成することにより、表構造を把握し、所望の記述形式に従った表記述を生成する文書変換方法。

【請求項４】請求項１乃至３のいずれかに記載の文書変換方法において、予め作成された図表の含まれる文書、図および表のうち少なくとも１つを除去した文書を生成する文書変換方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、コンピュータ装置上で動作する文書処理プログラムに係り、特に、ワープロ等で作成される文書について、文書の構造変換および記述形式の変換を行う文書処理方法に関する。

【０００２】

【従来の技術】ワープロの普及により文書の電子化が進み、過去に作成した文書を編集して新たな文書を作成するといった文書の再利用が可能となった。しかし、様々な機種のワープロが存在し、各機種がそれぞれ独自の文書記述形式を用いているため、異機種間での文書データの交換は困難だった。単純なテキストのみの形式の文書であれば、すべての機種で利用可能であるが、図表を含む文書やレイアウト指定を含む文書は交換／再生ができなかった。

【０００３】この問題を解決すべく、文書の論理構造を表現するための標準的な文書記述言語ＳＧＭＬ(ISO 8879, Information processing - Text and office systems -Standard Generalized Markup Language(SGML))が提案された。ＳＧＭＬでは、ＤＴＤ(Document Type Definition:文書型定義)により、文書の構造およびそれを構成する構造要素の集合を定義し、これに基づいて文書を記述する。文書を構成する構造要素は、タグで囲むことにより明示的に示す。例えば、「＜タイトル＞変換方法について＜／タイトル＞」という記述により、文書のタイトルが「変換方法について」であることを表現する。構造要素名（この例では、「タイトル」）を「＜」と「＞」で囲んだものを開始タグと呼び、「＜／」と「＞」で囲んだものを終了タグと呼ぶ。

【０００４】PostScriptデータのような、テキスト表現の記述形式に従ったデータであれば、文書中に図表を含めて記述してもかまわない。バイナリ形式の画像データ等は、文書中に直接記述することはできないので、画像データが格納されたファイルを「エンティティ宣言」という構文を用いて参照する。いずれの場合にも、図や表の記述形式を「ノテーション宣言」によって示す。

【０００５】また、文書のレイアウトに関するレイアウト情報は文書中には含まない。文書のレイアウト処理を行うシステムにおいて、構造要素とレイアウトを関連付けておくことにより、文書のレイアウトが行われる。従って、文書を作成する機器等に依存しない文書の作成が可能となり、図表を含んだ文書の再利用が可能となる。

【０００６】

【発明が解決しようとする課題】ＳＧＭＬが導入される以前にワープロ等で作成された文書も含めて文書の再利用を可能にするシステムを構築するためには、様々な機種のワープロ等によりそれぞれ異なる文書記述形式で記述された文書を、標準的なＳＧＭＬ文書に変換することが必要となる。さらに、文書の幅広い活用のためには、ＳＧＭＬ文書から他の形式への変換も必要となる。

【０００７】本発明の目的は、ＳＧＭＬ文書に限らない、ワープロ等で作成された図表を含む文書を、ユーザの文書作成／参照環境に合った形式の文書に変換する変換方法を提供することにある。

【０００８】

【課題を解決するための手段】ワープロ等で作成された文書中には、文書内容を表わす文字列の他、図表データを表わす文字列や、文字列および図表の表示に関するレイアウト情報を表わす特定の文字列が、独自に定められた文書記述形式により記述されている。そのようなワープロ文書に含まれる特定の文字列を、ＳＧＭＬのタグ表現に置き換えれば、形式的には、ＳＧＭＬの構文に従った文書を生成することができる。

【０００９】ところで、特開平７－１０５２１６号公報に記載の発明では、ＳＧＭＬ文書を入力文書として文書構造の解析を行った後、文書構造を構成する各構造要素に対応した処理を実行する手段を備えることにより、構造要素単位の文字列変換、および文書構造の構造変換を

行うための処理を、ユーザが容易に指定することのできる手段を提供している。そして、文書構造を辿りながら、各構造要素について指定された処理を行うことにより、ＳＧＭＬ文書の変換を実現している。従って、ワープロ文書から生成されるＳＧＭＬ記述の文書をＳＧＭＬ文書とみなすことにより、上記方式を用いた文書の変換が可能となる。

【００１０】また、文書中に図表が含まれる場合には、図データを画像ファイルとして抽出したり、表データを他のアプリケーションで利用可能な形式に変換して抽出するといったことも、必要となる。ワープロ文書をＳＧＭＬ記述に置き換えることにより、その中に含まれる図表データの先頭および末尾に必ず存在する特定の文字列が、図データあるいは表データを表すタグ名を持つタグに置き換えられる。図データを表すタグで囲まれた部分は図の構造要素とみなすことができる。これについて、その内容を別ファイルに出力する処理を行うことにより、図データを切り出すことができる。必要であれば、これをバイナリ形式に変換するなどして画像ファイルを生成する。同様に、表データを表すタグで囲まれた部分は表の構造要素とみなすことができる。表データについては、罫線位置等の表の構造に関する情報を示す特定の文字列もタグに置き換えられるため、表データ自身も罫線情報等の構造要素からなる表構造データとして表現される。従って、罫線情報等の各構造要素についても実行すべき処理を定義することができるため、表の構造に関するすべての情報を格納するテーブルを生成して、表の構造を把握することも容易である。また、構造変換処理と構造要素単位の文字列変換処理を行うことにより、ユーザの所望の記述形式に従った表データを生成することもできる。

【００１１】

【発明の実施の形態】以下、本発明の実施例を図面に基づいて説明する。

【００１２】図１は、本発明の文書変換方式を利用するシステム構成の一例として、ネットワークに接続されたコンピュータ上に文書の変換を行う文書変換処理プログラムを置いたシステム構成を示す。ネットワーク７に接続されたコンピュータ１は、ディスプレイ２と、キーボード等のデータ入力装置３と、ＣＰＵ４と、メモリ５と、データ入力装置３から入力される文書およびネットワーク７を介して他のコンピュータから取得した文書を保存するためのデータファイル６とから構成される。メモリ５には、文書変換処理プログラム５－１と、文書変換処理プログラム５－１から起動される共通形式化プログラム５－２と、文書変換処理プログラム５－１から起動される文書構造解析プログラム５－３と、文書構造解析プログラム５－３がＳＧＭＬ文書を読み込み、これを構造解析して生成する文書構造データを格納するための文書構造格納領域５－４と、文書変換処理プログラム５－１が図表の含まれるＳＧＭＬ文書から抽出する表データあるいは画像データのそれぞれを格納するための表構造格納領域５－５と画像格納領域５－６と、画像ファイルのデータ形式の変換を行う画像変換処理プログラム５－７が置かれる。

【００１３】図２は、文書変換処理の概要を示す。入力文書は、コンピュータ１上で作成した文書、フロッピーディスクやＣＤ－ＲＯＭ等の可搬型媒体に格納されている文書、あるいはネットワーク７を介して取得した文書とする。ユーザは、文書入力時に、出力文書の記述形式を指定する。文書変換処理では、まず、入力文書の記述形式を判定する。入力文書がＳＧＭＬ文書であれば、図中に示すような木構造状の文書構造データを生成し、この文書構造データに対して構造変換および記述形式変換を行うことにより、指定された記述の文書を生成する。ＳＧＭＬ文書以外の文書については、まず、ＳＧＭＬ記述の共通形式文書に変更する。ＳＧＭＬ記述に変更するための方法としては、ワープロ等で作成された文書中に含まれる、文字列のセンタリング等のレイアウトを指定する命令文をタグ表現に置き換えることが考えられる。このような方法により共通形式文書を生成し、これをＳＧＭＬ文書とみなすことにより、ＳＧＭＬ文書と同様に構造変換および記述形式変換を行う。文書に図表が含まれる場合には、図表以外の文字列の部分と図の部分と表の部分とに分かれた文書構造が生成されるため、図の部分のみを画像ファイルとして出力したり、表の部分のみを表データファイルとして出力することも可能である。

【００１４】図３は、文書変換処理プログラムのフローチャートを示す。ステップ３０１で、入力文書の記述形式の判定を行う。ワープロ等で作成される文書の多くは、文書データの先頭部分に記述形式が明示されているため、先頭部分を参照することにより、記述形式は容易に判定できる。判定の結果、入力文書がＳＧＭＬ文書でない場合には、ステップ３０３で、文書データをＳＧＭＬ記述の共通形式文書に変換する共通形式化処理を行う。図４に示すＬＡＴＥＸの構文に従って記述されたテキスト文書を入力文書の例として、ステップ３０３の共通形式化処理について詳細に説明する。ＬＡＴＥＸ文書は¥documentstyle{...}で始まり、文書内容を表わす文字列と、文書のレイアウトに関する命令文とから構成される。¥で始まる命令文（例えば、¥title）は、文書をレイアウトするための、配置、フォント、文字サイズといったレイアウト情報に関係付けられている。特殊な命令文（例えば、¥documentstyle{jreport}）を除いて、ＬＡＴＥＸでは、命令文の後に続く中かっこ（｛、｝）で囲まれた文字列に対し、指定のレイアウトが適用される。命令文および中かっこで囲まれていない文字列、すなわち命令文の施されていない文字列もあるが、そのような場合には、標準的な配置、フォント、文字サイズを用いたレイアウトが適用される。共通形式化処理では、

図４のような文書について、命令文の表現をタグの表現に置き換える。例えば、タイトル部分「¥title{ＯＤＡに基づいた…}」については、「¥title{」を<title>に置き換え、その後の文字列に続く「}」を</title>に置き換えることにより、「<title>ＯＤＡに基づいた…</title>」という記述を生成する。同様に、章タイトル部分「¥chapter{…}」、節タイトル部分「¥section{…}」を、それぞれ「<chapter>…</chapter>」、「<section>…</section>」に置き換える。章タイトルや節タイトルのあとに出現する命令文の施されていない文字列については、これを段落とみなし、段落を表すタグ<para>を文字列の先頭と末尾に追加する。また、文書自体にも、文書の開始、終了を表すタグ（<doc>）を、文書の先頭、末尾に追加する。このような処理を行うことにより、図５のような共通形式文書が得られる。

【００１５】次に、ステップ３０４では、入力されたＳＧＭＬ文書について、またはステップ３０３で生成された共通形式文書をＳＧＭＬ文書とみなし、文書構文解析を行い、木構造状の文書構造データを生成する。このステップ３０４の処理では、図５に示すような共通形式文書を入力文書として、図６に示すような文書構造データを生成する。３０５ステップ以降の処理は、３０４ステップで生成された文書構造データについて行い、文書構造に含まれる各構造要素に関する変換処理の指定と、それらの変換処理の実行は、特開平７－１０５２１６号記載の方式により行うものとする。

【００１６】ステップ３０５では、文書中に図が含まれるかどうかを判定し、図が含まれる場合には、ステップ３０６で、図の部分を画像データファイルとして抽出する画像データ生成処理を行う。ステップ３０７では、文書中に表が含まれるかどうかを判定し、表が含まれる場合には、ステップ３０８で、表の構造を解析し、指定された出力形式に応じた表の記述を生成する表データ生成処理を行う。

【００１７】ステップ３０９では、指定の形式の文書を出力するために、特定の構造要素を除去、および構造要素の順序を入れ替える等の文書構造データの構造変換を行う。例えば、図５のような共通形式文書をＳＧＭＬ文書に変換する場合には、共通形式文書に含まれるレイアウト情報は不要となるため、図６に示すような文書構造データからレイアウト情報に関する構造要素を除去し、必要に応じて構造要素名を変更し、図７に示す「報告書」のような、階層的な文書構造に変換する。

【００１８】ステップ３１０では、構造変換後の文書構造データについて、文字列を指定された記述形式に変更して出力することにより、指定の出力形式の文書を生成する。例えば、図７のような文書構造データを辿りながら、各構造要素の内容に応じて文字列出力を繰り返すことにより、図８のようなＳＧＭＬ文書を出力することができる。

【００１９】図９は、図３のステップ３０６に示した画像データ生成処理の流れを示す。ＳＧＭＬでは、文書に含まれる図データ（画像データ）の記述方法としては、データファイル６上に存在する画像データファイル（以下、画像ファイルとする。）のファイル名のみを文書中に記述する方法と、バイナリ形式の画像データをテキスト表現に変換したテキスト形式の画像データを文書中に記述する方法とがある。画像データ生成処理では、テキスト形式の画像データが文書中に記述されている文書について、文書中に埋め込まれた画像データを画像ファイルとして抽出し、その画像ファイル名を文書中に書き込む。よって、もともと画像ファイル名が文書中に記述されているものについては、画像データ生成処理の必要はない。以下、画像データ生成処理について詳細に説明する。文書中に含まれる画像データの先頭および末尾には、画像データの開始、終了を示す特定の文字列が存在する。従って、テキスト表現に変換されたビットマップ形式の画像データを含む文書を共通形式に変換すると、図１０のような記述が得られる。図を表すタグを<PICTURE>とすると、テキスト表現に変換されたビットマップデータは<PICTURE>タグと</PICTURE>タグとで囲まれる。このような記述を構造解析することにより、図１１のような、PICTURE構造要素の子として、ビットマップデータ文字列を持つ文書構造データが生成される。一般に、画像データの先頭には、画像データに関する情報（以下、画像ヘッダ情報とする。）として、データ格納形式が記述されており、従って、画像ヘッダ情報を読み取ることにより、データ格納形式は容易に得られる。そこで、ステップ３０６２では、テキスト表現に変換されたビットマップデータ文字列に含まれるヘッダ情報を読み取り、図のデータ格納形式を取得する。ステップ３０６４で、図データのみを格納するためのファイル名を生成する。ステップ３０６６で、図（PICTURE）の子であるビットマップデータ文字列を、ステップ３０６４で生成されたファイル名を持つテキストファイルとして出力する。ステップ３０６８で、図（PICTURE）の子（ビットマップデータ文字列）を、ステップ３０６４で生成されたファイル名に置き換える。ステップ３０７０では、ステップ３０６６で出力された図ファイルについてデータ形式の変換を行う。例えば、図１０のようなテキスト化されたビットマップデータについては、ステップ３０６６で出力されるテキストファイルについてバイナリ変換を行い、ビットマップファイルを生成する。さらに、必要に応じて、画像変換プログラム５－６を用いて、他の画像データ格納形式への変換を行う。

【００２０】図１２は、図３のステップ３０８に示した表データ生成処理の流れを示す。表データ生成処理の対象となる表の例を図１３に示す。図１３の表をＬＡＴＥＸで記述すると、図１４のように記述できる。以下、図１４の記述について説明する。最初の¥begin{tabular}

は表記述の開始を表す。それに続く{|c|c|c|}は表の行の属性を指定するパラメータで、一つの行が３つのセルからなり、それぞれのセル間を縦の罫線で区切り、各セルにおいて文字列をセンタリングすることを指定する。¥hlineや¥clineは、その位置に横の罫線を引くことを表し、¥hlineは行に含まれるすべてのセルに罫線を引き、¥cline{2-3}は罫線を引くセルの範囲をパラメータ（この例では、{2-3}）で指定することができる。また、&は表のセル間の区切り位置を、¥¥は改行を表す。最後の¥end{tabular}は、表記述の終了を表す。

【００２１】図１４に示すような表記述は、図３のステップ３０３に示す共通形式化処理により、図１５のような共通形式に変換される。図１５の記述について、図３のステップ３０４に示す文書構造解析処理を行うことにより、図１６のような文書構造データが生成される。表データ生成処理では、図１６のような文書構造データを対象に、表構造の把握と、所望の表記述を生成する処理を行う。まず、ステップ３０８２では、この文書構造データをもとに図１７に示すような表構造テーブルを生成し、表に含まれるすべてのセルに関する罫線情報およびセル中の文字列を、表構造テーブルに書き込んでいく。テーブル中の罫線情報には、セルの統合を検出するために必要な情報として、セルを囲む上下左右の罫線の有無とその位置座標、縦方向あるいは横方向に隣接するセルの統合について、その統合セル数が格納できるものとする。

【００２２】図１６のような文書構造データについては、表の罫線に関する構造要素hline,clineをもとに各セルの罫線情報を書き込んでいく。「hline」はその行に含まれるすべてのセルの下罫線を引き、それは同時に、その次の行に含まれるすべてのセルの上罫線となる。「cline」（例えば、属性「2-3」を持つとする。）は、その行の２、３番目のセルにのみ下罫線を引くため、その次の行についても２、３番目のセルにのみ上罫線が存在することになる。

【００２３】すべてのセルの罫線情報の書き込み終了後、これらの罫線情報をもとにセル間縦方向／横方向の統合を検出する。図１３の表を例にとると、１列目の左端のセルと２列目の左端のセルは縦方向に統合されている。すなわち、１列目の左端のセルは、縦方向の統合開始セルであり、これは、表構造テーブルにおいて、上罫線が存在し、かつ、下罫線が存在しないセルとして検出することができる。縦方向の統合開始セルを検出したら、次の列の、縦方向に隣り合うセルの罫線情報を参照し、もし、下罫線があれば、それを統合終了セルとし、下罫線がなければ、さらに縦方向の統合が続くものとみなす。縦方向のセルの統合は、隣り合うセルを順にたどって、下罫線の存在するセルにたどり着くまで続き、下罫線の存在するセルを統合終了セルとする。統合開始セルから統合終了セルまでのセルの数は、統合開始セルに関する罫線情報中の縦方向統合数として書き込む。横方向の統合についても同様に、セルの右罫線に着目することにより、統合数を検出し、横方向統合数として書き込む。ただし、ＬＡＴＥＸでは、セルの横方向の統合に関して、その統合数を命令文¥multicolumnのパラメータとして記述することができるため、統合数はパラメータから容易に得られる。図１８のような表は、ＬＡＴＥＸでは図１９のように記述される。

【００２４】ステップ３０８２の表構造テーブル生成が終了したら、ステップ３０８４で、図１６のような表構造を、出力形式に合わせた構造に変換する表構造変換を行う。例えば、ＨＴＭＬ形式の文書を出力する場合には、ＨＴＭＬの構文に合った、図２０のような構造に変換する。ステップ３０８６では、このような木構造をたどりながら、文字列を出力することにより、図２１のようなＨＴＭＬ記述を出力する。

【００２５】

【発明の効果】本発明によれば、ワープロ文書について、レイアウト情報を表す特定の文字列をＳＧＭＬのタグ表現に置き換えることにより、ＳＧＭＬ記述の共通形式の文書を生成する。これをＳＧＭＬ文書とみなして処理することにより、文書の記述形式の変更や、文書中に含まれる図表を取り出すといった文書の多様な変換が容易に行えるようになる。従って、機種を問わない文書の交換、および再利用が可能となる。

【図面の簡単な説明】

【図１】本発明のシステム構成図である。

【図２】本発明における処理の概要を示す図である。

【図３】文書変換処理プログラムのフローチャートである。

【図４】入力文書の例を示す図である。

【図５】共通形式文書の例を示す図である。

【図６】共通形式文書の文書構造を示す図である。

【図７】構造変換の例を示す図である。

【図８】出力文書の例を示す図である。

【図９】画像データ生成処理を示す図である。

【図１０】図の記述例を示す図である。

【図１１】図の構造例を示す図である。

【図１２】表データ生成処理を示す図である。

【図１３】表構造テーブルを示す図である。

【図１４】本発明が対象とする第１の表の例を示す図である。

【図１５】第１の表記述の例を示す図である。

【図１６】表記述の共通形式化例を示す図である。

【図１７】表構造の例を示す図である。

【図１８】本発明が対象とする第２の表の例を示す図である。

【図１９】第２の表記述の例を示す図である。

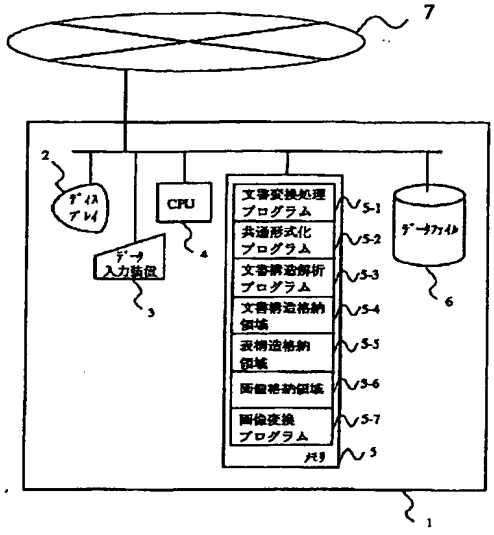
【図２０】表構造の変換例を示す図である。

【図２１】表記述の出力例を示す図である。

【符号の説明】
１…コンピュータ、２…ディスプレイ、３…データ入力＊　＊装置、４…ＣＰＵ、５…メモリ、６…データファイル、７…ネットワーク。

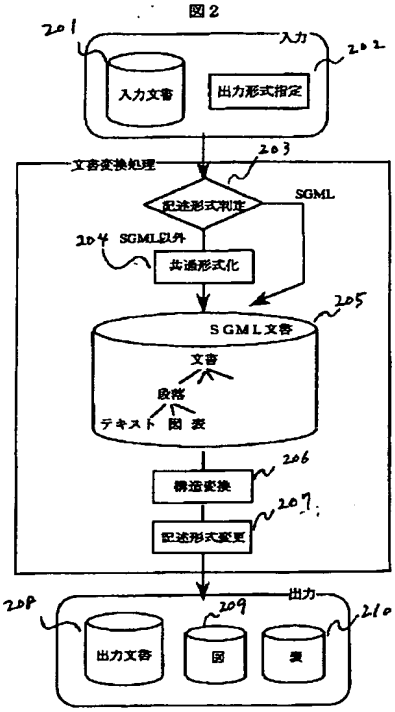
【図１】

図１



【図２】

図２



【図１０】

図１０

```
<PICTURE>
424d9602000000000003e0000002800
0000480000003200000001000100000
0000580200000000000000000000200
0000020000000000000ffffff00ffff
…
</PICTURE>
```

【図１４】

図１４

```
¥begin{tabular}{|c|c|c|}  ¥hline
blue & red    & violet ¥¥ ¥cline{2-3}
     & yellow & green  ¥¥ ¥hline
red  & white  & pink   ¥¥ ¥hline
¥end{tabular}
```

【図４】

図４

```
¥documentstyle{jreport}

¥title{ODAに基づいた…}
¥author{XXXX}
¥date{1996.2.15}

¥begin{document}
¥maketitle
¥chapter{はじめに}
1989年にISOで規格化されたODAでは，論理構造と…
…
¥chapter{割り付け処理における…}
¥section{従属接続子}
共通割り付け構造は，…
…
¥chapter{おわりに}
ODAに基づく…
…

¥end{document}
```

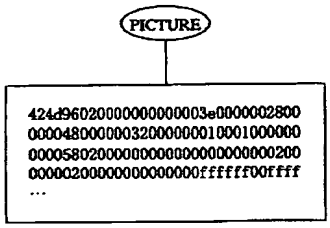
【図５】

図５

```
<doc>

<documentstyle>jreport</documentstyle>

<title>ODAに基づいた…</title>
<author>XXXX</author>
<date>1996.2.15</date>

<begin>document</begin>
<maketitle>
<chapter>はじめに</chapter>
<para>1989年にISOで規格化されたODAでは，論理構造と…
…
<chapter>割り付け処理における…</chapter>
<section> 従属接続子</section>
<para> 共通割り付け構造は…
…
<chapter>おわりに</chapter>
<para>ODAに基づく…
…

<end>document</end>

</doc>
```

【図１１】

図１１



【図１３】

図１３

blue	red	violet
	yellow	green
red	white	pink

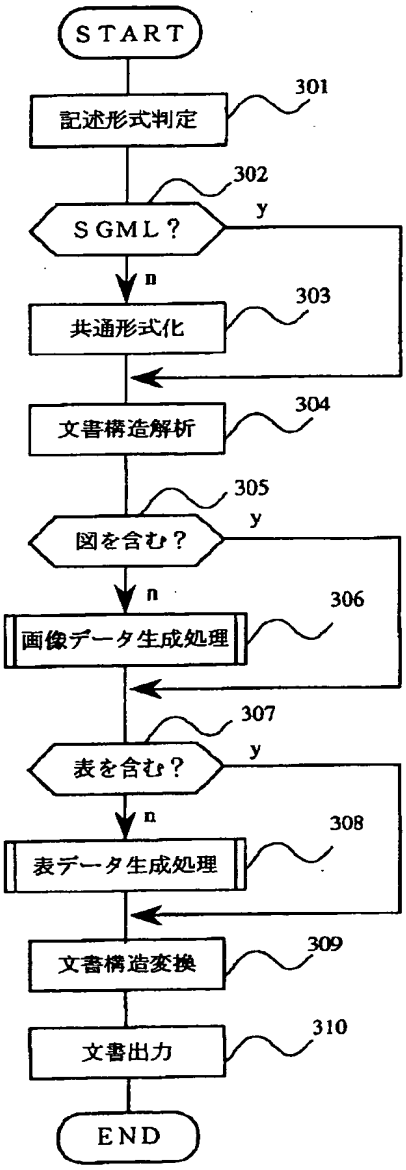
【図１８】

図１８

red	blue	
white	red	yellow
pink	violet	green

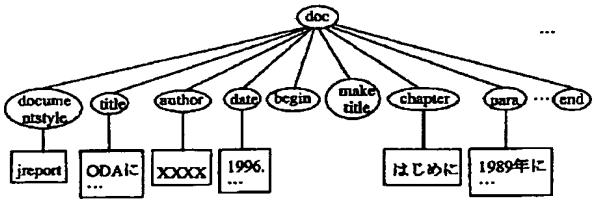
【図３】

図３



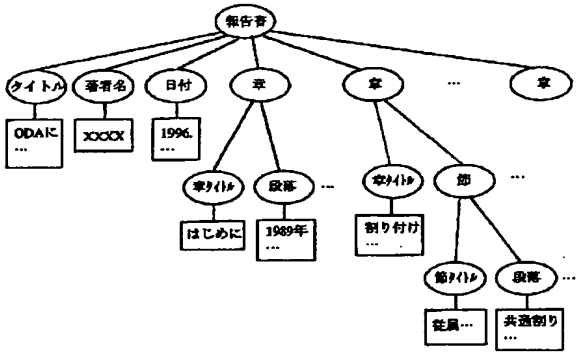
【図６】

図６



【図７】

図７



【図８】

図８

```
<!DOCTYPE 報告書 SYSTEM "報告書.DTD">

<報告書>

<タイトル>ODAに基づいた…</タイトル>
<著者名>XXXX</著者名>
<日付>1996.2.15</日付>

<章>はじめに
<段落>1989年にISOで規格化されたODAでは，論理構造と…
<段落>…
…
<章>割り付け処理における…
<節>従属接続子
<段落>共通割り付け構造は，…
…
<章>おわりに
<段落>ODAに基づく…
…

</報告書>
```

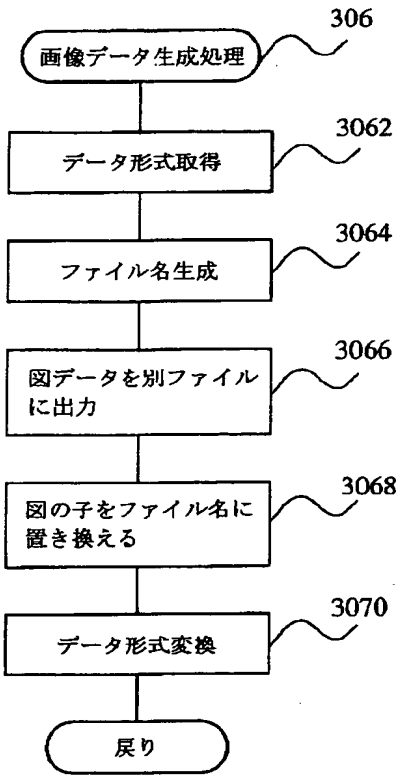
【図１９】

図１９

```
¥begin{tabular}{|c|c|c|}  ¥hline
red   & ¥multicolumn{2}{c|}{blue}  ¥¥ ¥hline
white & red    & yellow             ¥¥ ¥hline
pink  & violet & green              ¥¥ ¥hline
¥end{tabular}
```

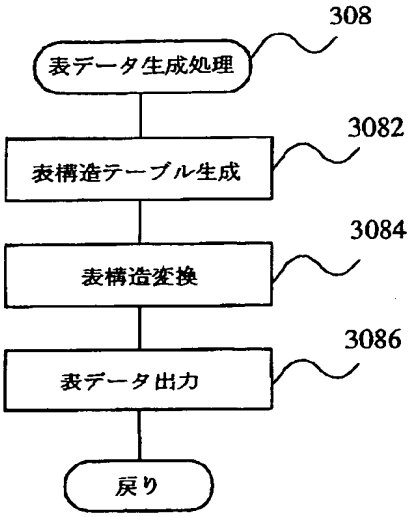
【図９】

図９



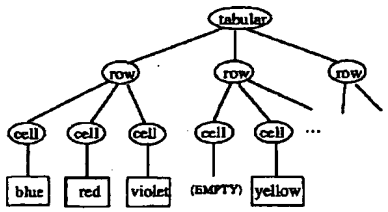
【図１２】

図１２



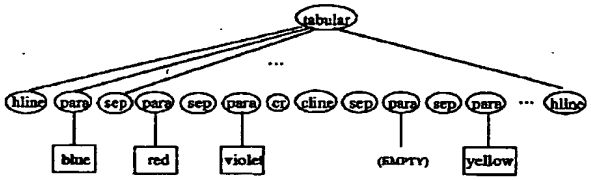
【図２０】

図２０



【図１６】

図１６



【図１５】

図１５

```
<tabular attr="|c|c|c|">
<hline>
<para>blue<sep><para>red<sep><para>violet<cr><cline attr="2-3">
<para><sep><para>yellow<sep><para>green<cr><hline>
<para>red<sep><para>white<sep><para>pink<cr><hline>
</tabular>
```

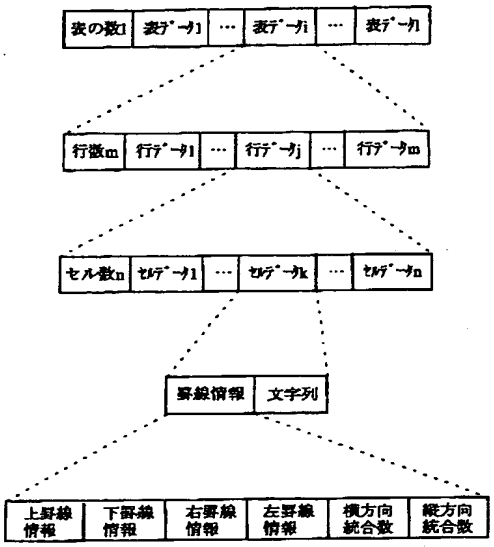
【図２１】

図２１

```
<TABLE BORDER>
<TR>
<TD ROWSPAN="2">blue</TD><TD>red</TD><TD>violet</TD>
</TR>
<TR>
<TD>yellow</TD><TD>green</TD>
</TR>
<TR>
<TD>red</TD><TD>white</TD><TD>pink</TD>
</TR>
</TABLE>
```

【図１７】

図１７


表の数l
表ﾃﾞｰﾀ1
…
表ﾃﾞｰﾀi
…
表ﾃﾞｰﾀl
行数m
行ﾃﾞｰﾀ1
…
行ﾃﾞｰﾀj
…
行ﾃﾞｰﾀm
セル数n
ｾﾙﾃﾞｰﾀ1
…
ｾﾙﾃﾞｰﾀk
…
ｾﾙﾃﾞｰﾀn
罫線情報
文字列
上罫線情報
下罫線情報
右罫線情報
左罫線情報
横方向統合数
縦方向統合数